Decentralized Control of a Collective of Autonomous Robotic Vehicles

David A. Schoenwald, John T. Feddema, and Fred J. Oppel

Intelligent Systems & Robotics Center
Sandia National Laboratories[1]
P.O. Box 5800
Albuquerque, NM 87185-1004
{daschoe,feddema,fjoppel}@sandia.gov

Abstract

In this paper, the performance of a group of autonomous vehicles tracking a prescribed goal is analyzed. The vehicles are considered to be ground-based unmanned robots acting as a group to maintain an unbroken communication network in a building or some other region. Vehicle interactions are modeled as a chain of interconnected systems. The stability of the entire collective as well as individual vehicles is studied using large-scale systems theory. Stability can be controlled via two key parameters: vehicle speed constant (maximum vehicle speed times sample time) and vehicle interaction gain. In addition to the stability analysis, simulation of a group of vehicles in a building with walls, doors, and other obstacles is studied with respect to maintaining a communication network among the vehicles at all times.

1. Introduction

As part of a project for DARPA's ITO Software for Distributed Robotics Program, Sandia National Laboratories is developing analysis and control software for coordinating hundreds to thousands of autonomous cooperative robotic agents performing military operations such as reconnaissance, surveillance and target acquisition; countermine and explosive ordnance disposal; force protection and physical security; and logistics support. Due to the nature of these applications, the control techniques must be distributed, and they must not rely on high bandwidth communication between agents. The goal of this work is to coordinate the behavior of a large number (10s to 100s to 1000s) of autonomous robotic agents performing various maneuvers. The algorithms to control these agents must be safe (provably stable and convergent), covert (low communication bandwidth), and fault tolerant (decentralized).

We have focused our effort on the task of surveillance wherein large numbers of vehicles from 20 to 1000 are dispersed around a facility. The goal is for these vehicles to autonomously create a distributed communication/navigation network that links a remote base station to multiple surveillance points. We have simulated in great detail the control of low numbers of vehicles (up to 20) navigating throughout a building. These simulations include detailed models of the radio frequency communication, infrared and ultrasound ranging with the environment and amongst vehicles, and the vehicle kinematics.

Techniques of large-scale systems theory are employed in analyzing the stability of these vehicles such as are described in [1]. More advanced techniques that use adaptive control [4] or variable structure control [5] are also being considered. The following sections describe the details on the stability analysis and the simulation results.

2. Stability Analysis for One-Dimensional Case

The subject of cooperative multiple autonomous vehicles has generated a great deal of interest in recent years due to the vision of these vehicles being able to perform tasks faster and more efficiently than an individual vehicle. Such tasks can include operations in hazardous or remote environments with the robot performing repetitive, dangerous, or information gathering duties. Recent work has taken many different approaches. The strategies employed are based on diverse fields such as artificial intelligence, game theory, biology, distributed control, and genetic algorithm[1]s. Because we are interested in proving convergence and stability of algorithms, we have been investigating using large-scale system control theory, as for example given in [1].

In complex large-scale systems, it is often desirable to break up a system into smaller strongly coupled systems that are controllable. If we can prove that the smaller systems are input/output reachable and controllable, then we can prove that the large-scale system is connectively controllable. Even the smaller scale systems may contain thousands of states, in which case, there exist techniques that can quickly determine whether the system is input/output reachable and structurally controllable. The analysis below shows some of the progress made in understanding how these techniques can be used in the design of large-scale distributed cooperative robotic vehicular systems.

We start by analyzing a simple one-dimensional problem in which a linear chain of interdependent vehicles is to spread out along a line as shown in Figure 1. The objective is to spread out evenly along the line using only information from the nearest neighbor. We had previously developed a robotic perimeter detection system that spread the vehicles around a perimeter using one-half the distance between the neighboring two vehicles as the goal point for each vehicle [3]. We were interested in finding out if one-half was a magic number and if we could prove that it provides a stable solution.

Assume that the vehicle's plant model is a simple integrator, and the commanded input is the desired velocity of the vehicle along the line. A feedback loop and a proportional gain K_p are used to control the vehicle's position. The desired position of each vehicle is one-half the sum of the position of the neighbors on each side. Figure 2 shows a block diagram of the control system. The formulation is in the discrete-time frequency domain, i.e. the z domain. Since we are interested in steady state analysis, we will make heavy use of the final value theorem, which states

$$\lim_{k\to\infty} f(kT) = \lim_{z\to 1}(1-z^{-1})F(z) \quad \textbf{(1)}$$

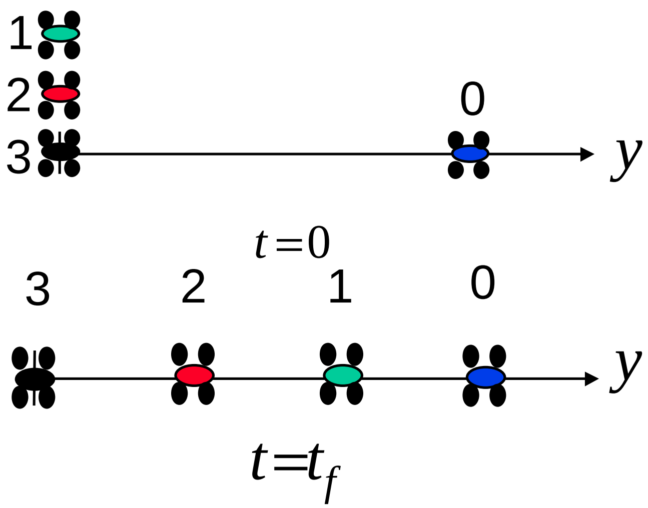

Figure 1. One-dimensional control problem. The top line is the initial state. The second line is the desired final state. The vehicles can only use their neighbors' position to reach the final goal state.

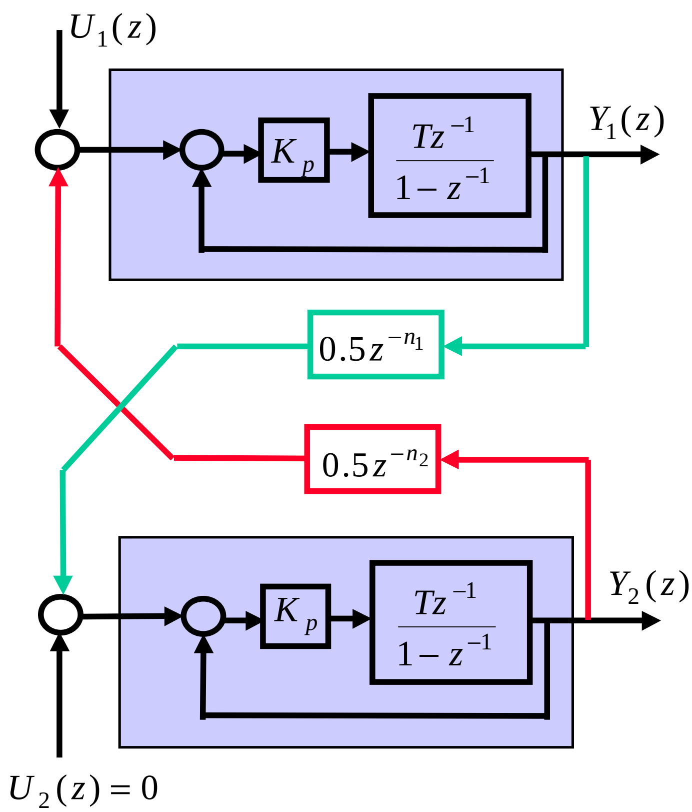

Figure 2. Control block diagram of two-vehicle interaction problem.

If we let $H_1(z)$ be the transfer function $Y_1(z)/U_1(z)$ and $H_2(z)$ be the transfer function $Y_2(z)/U_1(z)$ then we have

$$y_1^{ss} = \lim_{k\to\infty} y_1(kT) = \lim_{z\to 1} H_1(z)$$
$$y_2^{ss} = \lim_{k\to\infty} y_2(kT) = \lim_{z\to 1} H_2(z)$$

where we have used the fact that $U_1(z) = 1/(1-z^{-1})$, i.e., $u_1(kT)=1$ for all k. That is, the desired linear positioning behavior has been normalized to be between 0 and 1. The superscript "ss" refers to the steady state value. Carrying out the block diagram manipulation and algebra, one arrives at the formulas for the steady state position values of the two vehicles in terms of the given parameters

$$y_1^{ss} = \frac{1}{1-\lambda_1\lambda_2}$$

$$y_2^{ss} = \frac{\lambda_1}{1-\lambda_1\lambda_2}$$

where λ_1 and λ_2 are the interaction gains (which are 0.5 in Figure 2). It should be noted that both steady state positions are independent of the delays n_1 and n_2, the proportional gain K_p, and the sampling time delay T. Likewise, the formulas for the interaction gains, given the steady state positions, are

$$\lambda_1 = \frac{y_2^{ss}}{y_1^{ss}}$$
$$\lambda_2 = \frac{y_1^{ss}-1}{y_2^{ss}}$$

These formulas assume a stable configuration. We will analyze stability shortly. Now consider the three-vehicle case. Using the same analysis as above, we can arrive at the steady state positions (assuming stability) for the three vehicles as

$$y_1^{ss} = \frac{1-\lambda_{32}\lambda_{23}}{1-\lambda_{21}\lambda_{12}-\lambda_{32}\lambda_{23}}$$
$$y_2^{ss} = \frac{\lambda_{12}}{1-\lambda_{21}\lambda_{12}-\lambda_{32}\lambda_{23}}$$
$$y_3^{ss} = \frac{\lambda_{12}\lambda_{23}}{1-\lambda_{21}\lambda_{12}-\lambda_{32}\lambda_{23}}$$

where again only the vehicle interaction gains affect the steady state position values. To solve the inverse problem, i.e., the vehicle interaction gains given the steady state positions, we must solve an under-determined system of nonlinear equations with 3 equations and 4 unknowns. This can be done using a nonlinear root finding algorithm such as employed in the MATLAB (trademark of The MathWorks, Inc.) routine, fsolve.

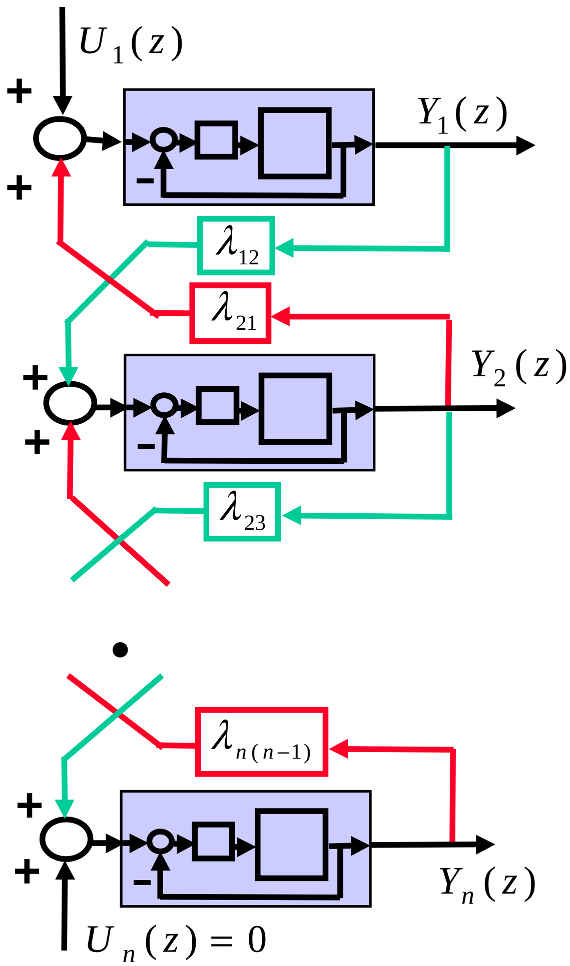

Figure 3. N-vehicle interaction problem.

To generalize the above for the N vehicle interaction problem (shown in Figure 3), we formulate a set of linear equations based on the algebra of the transfer function manipulation. Note that

$$
\begin{aligned}
y_1^{ss} &= 1 + \lambda_{21} y_2^{ss} \\
y_i^{ss} &= \lambda_{i-1,i} y_{i-1}^{ss} + \lambda_{i+1,i} y_{i+1}^{ss} \\
y_N^{ss} &= \lambda_{N-1,N} y_{N-1}^{ss}
\end{aligned}
$$

where i is an integer such that $i \in [2, N-1]$.

This results in the system of linear equations

$$
\begin{bmatrix} y_1^{ss} \\ \vdots \\ y_i^{ss} \\ \vdots \\ y_N^{ss} \end{bmatrix} = \begin{bmatrix} 0 & \lambda_{21} & 0 & \cdots & 0 \\ \lambda_{12} & 0 & \lambda_{32} & \cdots & 0 \\ 0 & \lambda_{i-1,i} & 0 & \lambda_{i+1,i} & 0 \\ \vdots & \ddots & \ddots & \ddots & \lambda_{N,N-1} \\ 0 & \cdots & 0 & \lambda_{N-1,N} & 0 \end{bmatrix} \begin{bmatrix} y_1^{ss} \\ \vdots \\ y_i^{ss} \\ \vdots \\ y_N^{ss} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}
$$

which can be reformulated in the familiar Ax=b form

$$
\begin{bmatrix} 1 & -\lambda_{21} & 0 & \cdots & 0 \\ -\lambda_{12} & 1 & -\lambda_{32} & \cdots & 0 \\ 0 & -\lambda_{i-1,i} & 1 & -\lambda_{i+1,i} & 0 \\ \vdots & \ddots & \ddots & \ddots & -\lambda_{N,N-1} \\ 0 & \cdots & 0 & -\lambda_{N-1,N} & 1 \end{bmatrix} \begin{bmatrix} y_1^{ss} \\ \vdots \\ y_i^{ss} \\ \vdots \\ y_N^{ss} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \quad \textbf{(2)}
$$

Thus, given the steady state position values, the required interaction gains can be solved for from a system of linear equations. The solution will not be unique, hence many different sets of interaction gains can result in the same steady state position values of the vehicles. Likewise, the inverse problem can be solved to determine the interaction gains given a set of desired steady state vehicle position values. We obtain

$$
\begin{bmatrix} 0 & y_2^{ss} & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ y_1^{ss} & 0 & 0 & y_3^{ss} & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & y_2^{ss} & 0 & 0 & y_4^{ss} & 0 & \cdots & 0 \\ 0 & & & & \cdots & & & & 0 \\ 0 & \cdots & y_{i-1}^{ss} & 0 & 0 & y_{i+1}^{ss} & 0 & \cdots & 0 \\ 0 & & & & \cdots & & & & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & y_{N-1}^{ss} & 0 \end{bmatrix} \begin{bmatrix} \lambda_{12} \\ \lambda_{21} \\ \lambda_{23} \\ \vdots \\ \lambda_{i-1,i} \\ \lambda_{i,i-1} \\ \vdots \\ \lambda_{N-1,N} \\ \lambda_{N,N-1} \end{bmatrix} = \begin{bmatrix} y_1^{ss}-1 \\ y_2^{ss} \\ y_3^{ss} \\ \vdots \\ y_i^{ss} \\ \vdots \\ y_N^{ss} \end{bmatrix}
$$

(3)

In this case, Ax=b is an under-determined system (more unknowns than equations, 2(N-1)>N for N>2) which can be solved using QR factorization such as with the MATLAB backslash (\) operator. Alternatively, this can be solved as a constrained linear minimum norm problem:

$$
\min_x \frac{1}{2} \|Ax - b\| \quad s.t. \quad x \geq 0, \quad Ax = b, \quad \varepsilon \leq x \leq 1 - \varepsilon
$$

where the first constraint rejects negative interaction gains, the second constraint forces (3) to be solved exactly, and the third constraint rejects zero and unity interaction gains, that is, ε is a small parameter greater than zero. The advantage of formulating (3) as a constrained least squares problem is that we can eliminate nonzero interaction gains from the set of possible solutions. Since zero interaction gains correspond to a vehicle not utilizing information of its nearest neighbors, it is best to look at nonzero interaction gains.

We now turn to the problem of analyzing stability of the N vehicle interaction problem. For this, a reformulation of the vehicle dynamics into discrete-time state space is helpful. The purpose of this analysis is to determine conditions for asymptotic stability of vehicle positions with respect to the interaction gains λ and vehicle speed time constant K_pT where T is the sample period. The following time-domain equations are derived from Fig. 4:

$$
\begin{aligned}
y_1(k+1) &= (1 - K_pT) y_1(k) + K_pT\lambda_{21} y_2(k-1) + K_pTu_1(k) \\
y_i(k+1) &= (1 - K_pT) y_i(k) + K_pT\lambda_{i-1,i} y_{i-1}(k-1) + K_pT\lambda_{i+1,i} y_{i+1}(k-1), \quad 2 \leq i \leq N-1 \\
y_N(k+1) &= (1 - K_pT) y_N(k) + K_pT\lambda_{N-1,N} y_{N-1}(k-1)
\end{aligned}
$$

(4)

where it is assumed that $u_N(k)=0$ and the delay between position interaction information is one sampling delay. To solve these equations, initialize by setting $y_i(0)= y_i(1)=0$ for i between 1 and N (note that initial vehicle positions do not have to start at 0, but this is the normal case). Then start the difference equation solver at k=1 (i.e. compute $y_i(2)$). For the stability analysis, we note that we can put (4) into a state space description. We break the analysis into two cases.

Case I: If all the interaction delays =0, i.e. $n_{ij}=0$ then we get the following state space description:

$$\begin{bmatrix} y_1(k+1) \\ y_2(k+1) \\ \vdots \\ y_i(k+1) \\ \vdots \\ y_{N-1}(k+1) \\ y_N(k+1) \end{bmatrix} =$$

$$\begin{bmatrix} 1-K_pT & K_pT\lambda_{21} & & & & & 0 \\ K_pT\lambda_{12} & 1-K_pT & K_pT\lambda_{32} & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & K_pT\lambda_{i-1,i} & 1-K_pT & K_pT\lambda_{i+1,i} & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & K_pT\lambda_{N-2,N-1} & 1-K_pT & K_pT\lambda_{N,N-1} \\ 0 & & & & & K_pT\lambda_{N-1,N} & 1-K_pT \end{bmatrix} \bullet$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \\ \vdots \\ y_i(k) \\ \vdots \\ y_{N-1}(k) \\ y_N(k) \end{bmatrix} + \begin{bmatrix} K_pT \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} u_1(k)$$

which is in the form: $y(k+1) = Ay(k) + Bu_1(k)$. The eigenvalues can easily be solved for in any of a number of software packages including MATLAB. For stability, we look at the maximum absolute value of all the eigenvalues of A, which is a real NxN matrix. If this is inside the unit circle (less than unity magnitude) then we have asymptotic stability of the vehicle positions. Otherwise, we do not have a stable vehicle configuration. Note that the A matrix above is in tridiagonal form. For the special case of all the interaction gains $\lambda_{ij} = \lambda$, the elements of each diagonal are equal. There is a special formula (p. 59 of [2]) for the eigenvalues of A in this case which is

$$eig(A) = 1 - K_pT + 2K_pT\lambda\cos\left(\frac{i\pi}{N+1}\right), i = 1,\cdots,N \qquad \textbf{(5)}$$

All the eigenvalues in (5) will be real. Figure 4 illustrates the stability region for this case. The dark zone represents stable combinations of interaction gain λ and K_pT (proportional control gain multiplied by the sampling period). The white zone represents unstable combinations of λ and K_pT. We refer to this as a stability "house" due to the shape of the stable zone. The size of this house varies only with N. The plot shown is for N=2. As N is increased, the house gets smaller in width but maintains the same height and shape. Figure 5 shows the stability region for N=10000. From the formula in (5), we can see that as $N \rightarrow \infty$ the cosine term becomes unity. This implies that λ must stay between –0.5 and 0.5 for K_pT less than one in order to maintain stability. For K_pT greater than one, the admissible λ values taper off parabolically (the sloped "roof") until K_pT =2. Computer simulations of (4) agreed with these stability results.

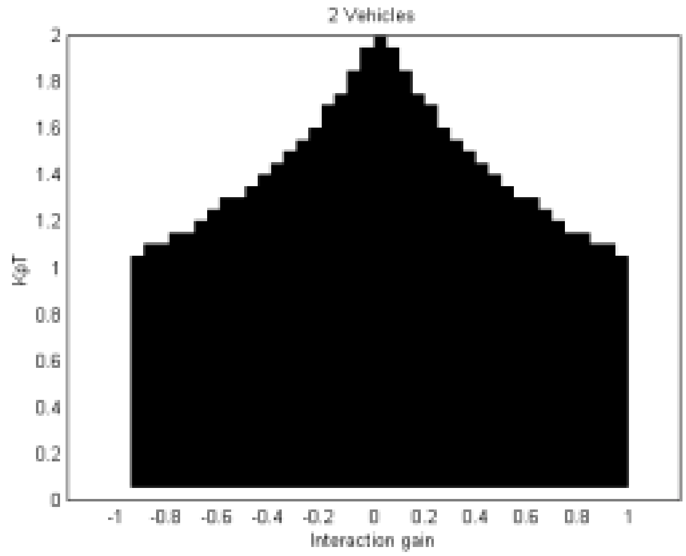

Figure 4. Stability region for the N=2 vehicle case.

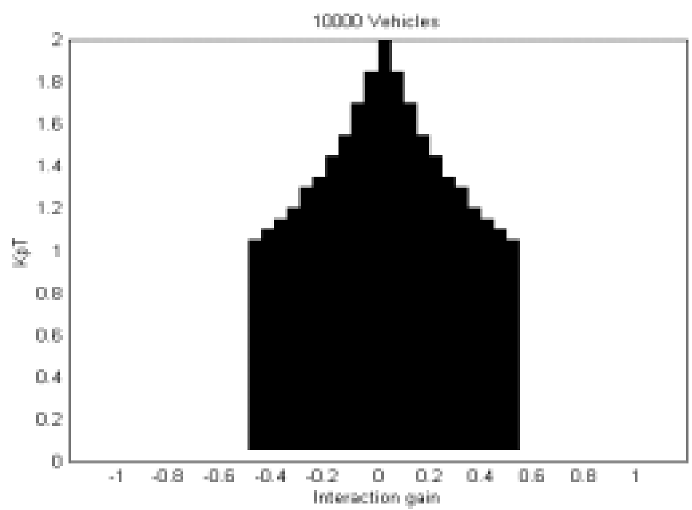

Figure 5. Stability region for the N=10000 vehicle case.

Case II: If all the interaction delays =1, i.e. $n_{ij}=1$, then we get a more complex state space description:

$$\begin{bmatrix} y_1(k+1) \\ \vdots \\ y_i(k+1) \\ \vdots \\ y_N(k+1) \\ y_1(k) \\ \vdots \\ y_N(k) \end{bmatrix} = \begin{bmatrix} 1-K_pT & & & 0 & 0 & K_pT\lambda_{21} & & \\ & \ddots & & & \ddots & \ddots & \ddots & \\ & & 1-K_pT & & & K_pT\lambda_{i-1,i} & 0 & K_pT\lambda_{i+1,i} \\ & & & \ddots & & & \ddots & \ddots \\ 0 & & & 1-K_pT & & & K_pT\lambda_{N-1,N} & 0 \\ & I_{NxN} & & & & 0_{NxN} & & \end{bmatrix} \bullet$$

$$\begin{bmatrix} y_1(k) \\ \vdots \\ y_i(k) \\ \vdots \\ y_N(k) \\ y_1(k-1) \\ \vdots \\ y_N(k-1) \end{bmatrix} + \begin{bmatrix} K_pT \\ 0 \\ \\ \\ \vdots \\ \\ \\ 0 \end{bmatrix} u_1(k)$$

The above still fits the $y(k+1)=Ay(k)+Bu_1(k)$ formulation. Note now that A is a 2Nx2N matrix. It is also no longer a tridiagonal matrix. There is no simple formula for the eigenvalues of A in this case even if all $\lambda_{ij} = \lambda$. The eigenvalues can still be solved for using standard linear algebra software, but this becomes numerically unreliable for large N. However, if a software package has techniques for handling large sparse matrices (as does MATLAB) then it becomes more tractable. In the above description, only 4N-2 elements of A are nonzero in general out of $4N^2$ total elements. Thus for large N, the A matrix is sparse. Though a formula is lacking, computer simulation of (4) and solving for the maximum absolute eigenvalue of A above resulted in exactly the same stability regions with respect to λ and K_pT . In other words, the delay in interaction gains between vehicles did not affect the stability of the vehicle positions to any appreciable degree.

A more specific case can be studied in which all forward λ_{ij}'s are equal (i.e. $\lambda_{12} = \lambda_{23} = \ldots = \lambda_F$) and all backward λ_{ij}'s are equal (i.e. $\lambda_{21} = \lambda_{32} = \ldots = \lambda_B$). In this case the stability region has a three-dimensional structure, K_pT vs. λ_F vs. λ_B. Numerical simulation of this case revealed that for various fixed K_pT contours from 1 to 2, the stability region for λ_F and λ_B looked like a $1/\lambda$ surface that increased in size as K_pT decreased from 2 down to 1. This is intuitive because we expect the range of λ's for stability to shrink as the speed gain is increased. This is essentially a three dimensional version of the "roof" of the stability house in Figs. 5 and 6.

Several conclusions can be drawn from the stability analysis. First, asymptotic stability of vehicle positions depends on vehicle responsiveness K_p, communication sampling period T, and vehicle interaction gain λ. If the vehicle is too fast (large K_p) or the sample period is too long (large T) then the vehicles will go unstable. There is a dependence on interaction gain for stability as well. Second, the interaction gains can be used to bunch the vehicles closer together or spread them out. Third, the stability region shrinks as the number of vehicles, N, increases but only to a defined limit. Finally, we can give a two step process for placing the vehicles into an arbitrary position. First, solve Eq. (3) for the λ_{ij}'s necessary to achieve these vehicle positions. Then, use the above stability analysis (the stability "house") to determine the upper limits for K_pT to maintain stability.

3. Simulation Example

The multiple-vehicle problem in planar space is essentially a generalization of the above analysis. But when obstacles such as walls and other vehicles as well as the need for communication between vehicles are taken into account, the ability to analytically solve the problem becomes very difficult. Thus, we implemented a study of multiple vehicles under such constraints in a simulation environment developed at Sandia National Laboratories called Umbra [6]. Umbra enables the simulation of multiple autonomous agents with a variety of physical phenomena such as RF communications, interactions with solid objects (i.e. collisions), ultrasound communication, IR detection of objects, vehicle physics, terrain descriptions, and other phenomena the user wishes to study. All of these physical attributes can be simulated simultaneously with a graphical visualization that allows the monitoring of the vehicles' performance over the terrain.

Such a simulation was implemented for the case of multiple, small, wheeled vehicles traversing a single floor in a building with multiple corridors, rooms, and entrances. The vehicles are modeled after the vehicles that will be used in the hardware tests. Each vehicle contains 4 IR sensors for detecting objects between 0.15m and 0.46m on all 4 sides of itself (see Figure 6). The vehicles also contain RF communication devices to be able to converse with other vehicles within a 30m line of sight range or roughly 10m through walls. They also have ultrasound capability to measure the distance between them provided they are within 10m of each other and in line of sight range. The vehicle physics are quite simple and proved adequate on a smooth surface. The building model was generated as a CAD model and contains several connected hallways as well as a multitude of variable size rooms. The control algorithms for the vehicles must avoid contact with walls and other vehicles. Beyond that, the control goals can vary depending on the motives of the operator. For instance, the vehicles can spread out to provide maximum coverage of the building or they can stay within a prescribed area, or they can maintain a particular formation. Note that a strict mathematical model of this situation is intractable. This is due to both discrete event-based as well as dynamic physics with very complicated interactions. Thus, the simulation shows stability in a qualitative fashion rather than strictly mathematical. However, future work will focus on demonstrating that these control algorithms are robust to modeling uncertainty.

The restriction that vehicles can't run into walls, doors, or each other essential ensures they remain inside the building. This is accomplished via rules that use the IR sensors to follow walls down a hallway. This will enable the vehicles to move throughout the building, though not necessarily in any prescribed fashion. Further restrictions on the vehicles involve the maintenance of a continuous RF communication network. This requires that vehicles stay within 30m of each other or less if line of sight (LOS) is lost (i.e. they may have to stay at a wall junction to maintain LOS). A more stringent condition is the ability for each vehicle to know its absolute (x,y) position with respect to some global coordinate system. This requires triangulation from two known vehicles using ultrasound as a distance measurement. This implies that at least two vehicles must remain in fixed known locations until the other vehicles can triangulate off of them. There are a number of techniques to accomplish this that were investigated in Umbra. These include the law of cosines triangulation, steepest descent triangulation, and conjugate gradient triangulation. All had advantages and disadvantages depending on the number of vehicles and the on-board processing power and memory. Finally, there is the constraint that the vehicles spread out and "cover" the building uniformly. A gradient-based scheme was used to repel the vehicles from each other to diffuse through the building while the aforementioned constraints keep them close enough to communication and compute absolute position. There are a variety of strategies for keeping the vehicles in contact with each other, both RF and ultrasound, to keep the communication loop intact. But the gradient-based scheme worked well because it minimized the number of vehicles needed to maintain the RF loop while still spreading the vehicles throughout the building.

It should be noted that the primary classes of vehicle maneuvers fall into 4 categories: 1) dispersion (diffusing throughout a space), 2) clustering (coming together to surround a target), 3) following (fairly linear progression through a space), and 4) orbiting (circular motion around a target). One technique we are investigating for coordination of these 4 maneuver types is sliding mode control. The sliding mode controller will switch between each category of vehicle maneuvers according to what is best to achieve overall stability and satisfaction of the target goals. Some preliminary work we have done in this area appears in [3].

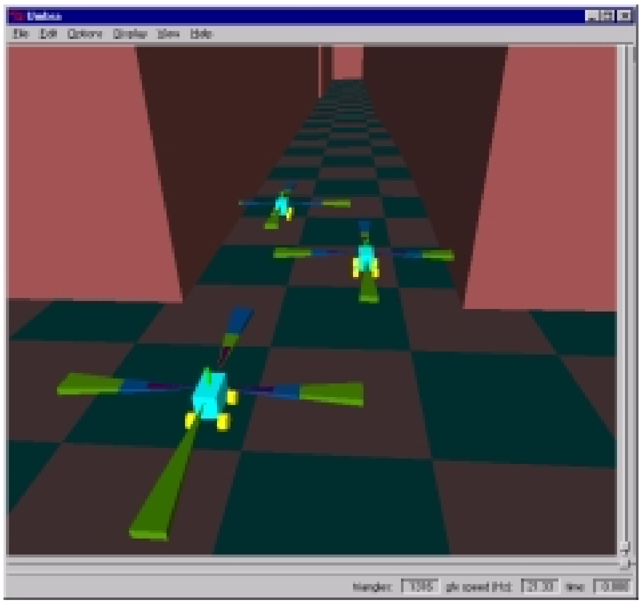

Figure 6. Detailed simulation of multiple vehicles navigating a building. The protruding green and blue cones represent the 4 IR proximity sensors.

Finally, some additional simulation in MATLAB shows a group of vehicles dispersing in an enclosed region. Figure 7 illustrates a group of 20 vehicles that starts in a tightly clustered position. They are tasked with the goal of spreading out uniformly in a room with walls determined by the boundaries of the graph. The results in Figure 7 show that the vehicles have spread out through the room fairly uniformly using a gradient-based scheme.

4. Conclusions

Work presented in this paper studies the stability problem of a multitude of autonomous robotic vehicles cooperating towards a prescribed goal. The analysis utilizes large-scale system theory and the control strategy is primarily decentralized to reduce communication overhead while maintaining a considerable amount of control authority at the vehicle level. This provides a degree of robustness should some of the vehicles fail. The analysis focuses on the linear case of vehicle motion and shows the stability regions. The simulation looks at a more complex situation with obstacles, RF communication, ultrasound position triangulation, and IR for obstacle avoidance. Hardware demonstrations of these vehicles and control strategies are in progress. We have built 20 low-cost robotic vehicle platforms, which contain the necessary processing and sensing to navigate and traverse a building. Currently, we are implementing many of the algorithms described above on these 20 vehicles. The goal of the simulation and the hardware tasks is to demonstrate that a cooperative group of robotic vehicles can form a communication/navigation network, and that this network could be applied to a surveillance task.

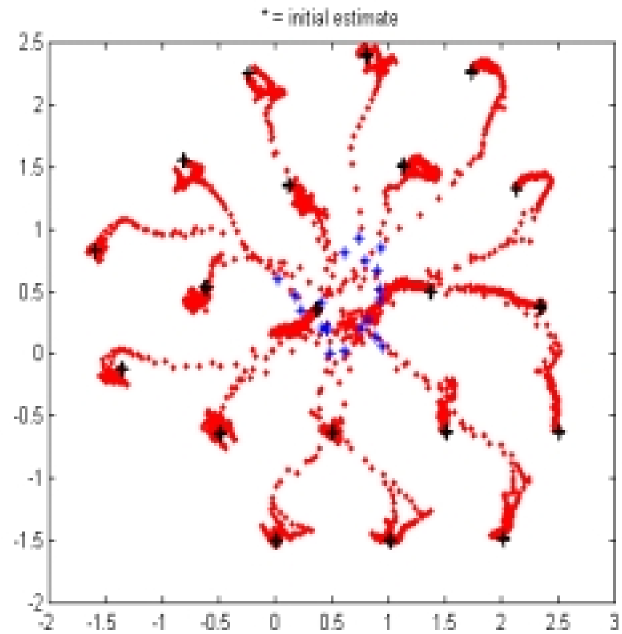

Figure 7. Plot of 20 vehicles' trajectories started from a clustered position with the goal of spreading out uniformly through the space (indicate initial position and + indicate final position).*

5. References

[1] D. D. Siljak, *Decentralized Control of Complex Systems*, Academic Press, 1991.
[2] G. D. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, Third Edition, Oxford University Press, 1985.
[3] J.T. Feddema, R.D. Robinett, and B.J. Driessen, "Designing stable finite state machine behaviors using phase plane analysis and variable structure control," *Proc. 1998 International Conf. on Robotics and Automation,* May 1998.
[4] Z. Wang, F. Khorrami, and Z. P. Jiang, "Adaptive Robust Decentralized Control for Interconnected Large-scale Nonlinear Systems," 2000 American Control Conference, Chicago, IL, pp. 347-351, June 2000.
[5] G. P. Matthews and R. A. DeCarlo, "Decentralized Tracking for a Class of Interconnected Nonlinear Systems Using Variable Structure Control," *Automatica*, vol. 24, no.2, pp. 187-193, 1988.
[6] D. E. Small, E. J. Gottlieb, K. Edlund, and C. Slutter, "A Design Patterns Analysis of the Umbra Simulation Framework," Sandia Report: SAND2000-2380, October 2000.

---

[1] Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94-AL85000. This research is partially funded by the Information Technology Office of the Defense Advanced Research Projects Agency as MIPR # 99H855 and Proposal # 096990208 titled "Analysis and Control Software for Distributed Cooperative Systems."